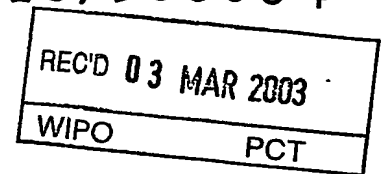


Rec'd PCT/PTO 24 JUN 2004

PCT/IB 02/05481

BUNDESREPUBLIK DEUTSCHLAND 24. 02. 03

10/500064



**Prioritätsbescheinigung über die Einreichung  
einer Patentanmeldung**

**Aktenzeichen:** 101 64 422.1

**Anmeldetag:** 29. Dezember 2001

**Anmelder/Inhaber:** Philips Corporate Intellectual Property  
GmbH, Hamburg/DE

**Bezeichnung:** Verfahren und Anordnung zum Beschreiben von NV-  
Memories in einer Controller-Architektur sowie ein  
entsprechendes Computerprogrammprodukt und ein  
entsprechendes computerlesbares Speichermedium

**IPC:** G 11 C, G 06 K, G 06 F

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ur-  
sprünglichen Unterlagen dieser Patentanmeldung.

München, den 30. Januar 2003  
Deutsches Patent- und Markenamt  
Der Präsident  
Im Auftrag

Faust

**PRIORITY  
DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

**BEST AVAILABLE COPY**

ZUSAMMENFASSUNG

Verfahren und Anordnung zum Beschreiben von NV-Memories in einer Controller-Architektur sowie ein entsprechendes Computerprogrammprodukt und ein entsprechendes computerlesbares Speichermedium.

- 5 Die Erfindung beschreibt ein Verfahren und eine Anordnung zum Beschreiben von NV-Memories in einer Controller-Architektur sowie ein entsprechendes Computerprogrammprodukt und ein entsprechendes computerlesbares Speichermedium, die insbesondere genutzt werden können, um Schreib- bzw. Programmiervorgänge in NV-Code-Memories von Mikrocontrollern, wie beispielsweise Smartcard-Controllern, zu beschleunigen.

10

Das Verfahren besteht in einer Erweiterung des Befehlssatzes des Controllers um sog. MOVCWR (move code write)-Instruktionen, die es ermöglichen, ein definiertes Datenwort (Byte) an eine definierte Zieladresse innerhalb eines NV-Code-Memories zu schreiben. Das Datenwort (Byte) wird hierbei an die korrekte Position des Cache-Pageregisters

- 15 des jeweiligen NV-Memories geschrieben und die Pageadreß-Pointerregister des Memories mit der zugehörigen Pageadresse aktualisiert. Wenn eine MMU (Memory Management Unit) vorhanden ist, geschieht dieses MOVCWR-Schreiben in das Cache-Pageregister, wie das MOVC-Lesen bzw. der Code-Fetch, unter Kontrolle dieser MMU.

## BESCHREIBUNG

Verfahren und Anordnung zum Beschreiben von NV-Memories in einer Controller-Architektur sowie ein entsprechendes Computerprogrammprodukt und ein entsprechendes computerlesbares Speichermedium.

- 5 Die Erfindung betrifft ein Verfahren und eine Anordnung zum Beschreiben von NV-Memories in einer Controller-Architektur sowie ein entsprechendes Computerprogrammprodukt und ein entsprechendes computerlesbares Speichermedium, die insbesondere genutzt werden können, um Schreib- bzw. Programmiervorgänge in NV-Code-Memories von Mikrocontrollern, wie beispielsweise Smartcard-Controllern, zu beschleunigen.

- 10 Die Entwicklung der Mikroelektronik in den 70er-Jahren ermöglichte es, kleine Computer im Kreditkartenformat ohne Benutzungsschnittstelle herzustellen. Solche Computer werden als Smartcards bezeichnet. In einer Smartcard sind Datenspeicher und arithmetisch-logische Einheiten in einem einzigen Chip von wenigen Quadratmillimetern Größe integriert.
- 15 Smartcards werden insbesondere als Telefonkarten, GSM-SIM-Karten, im Bankbereich und im Gesundheitswesen eingesetzt. Die Smartcard ist damit zur allgegenwärtigen Rechenplattform geworden.

- Smartcards werden derzeit vornehmlich als sicherer Aufbewahrungsort für geheime Daten und als sichere Ausführungsplattform für kryptographische Algorithmen betrachtet. Die Annahme einer relativ hohen Sicherheit der Daten und Algorithmen auf der Karte liegt im Hardwareaufbau der Karte und den nach außen geführten Schnittstellen begründet. Die Karte stellt sich nach außen als "Black Box" dar, deren Funktionalität nur über eine wohldefinierte Hardware- und Softwareschnittstelle in Anspruch genommen werden kann, und
- 25 die bestimmte Sicherheitspolices erzwingen kann. Zum einen kann der Zugriff auf Daten an bestimmte Bedingungen geknüpft werden. Kritische Daten, wie zum Beispiel geheime Schlüssel eines Public-Key-Verfahrens, können dem Zugriff von außen sogar völlig entzogen werden. Zum anderen ist eine Smartcard in der Lage, Algorithmen auszuführen, ohne daß die Ausführung der einzelnen Operationen von außen beobachtet werden kann. Die
- 30 Algorithmen selbst können auf der Karte vor Veränderung und Auslesen geschützt werden.

Im objektorientierten Sinn läßt sich die Smartcard als abstrakter Datentyp auffassen, der über eine wohldefinierte Schnittstelle verfügt, ein spezifiziertes Verhalten aufweist und selbst in der Lage ist, die Einhaltung bestimmter Integritätsbedingungen bezüglich seines Zustandes sicherzustellen.

5

Es gibt im Wesentlichen zwei verschiedene Typen von Smartcards. Speicherkarten besitzen lediglich eine serielle Schnittstelle, eine Adressierungs- und Sicherheitslogik und ROM- und EEPROM-Speicher. Diese Karten besitzen nur eingeschränkte Funktionalität und dienen einer spezifischen Anwendung. Dafür sind sie besonders billig herzustellen. Als

10 Mikroprozessorkarten hergestellte Smartcards stellen im Prinzip einen vollständigen Universalrechner dar.

Der Herstellungs- und Auslieferungsprozeß für Chipkarten gliedert sich in folgende Phasen:

- 15
- Herstellen des Halbleiters,
  - Einbetten des Halbleiters,
  - Bedrucken der Karte,
  - Personalisierung der Karte,
  - Ausgeben der Karte.

20

Im Allgemeinen wird jede Phase von einer auf die jeweilige Arbeit spezialisierten Firma durchgeführt. Beim Herstellen der Halbleiter ist insbesondere bei Karten mit festverdrahteter Sicherheitslogik auf eine gute betriebsinterne Sicherheit zu achten. Damit vom Hersteller ein korrekter Endtest durchgeführt werden kann, muß der komplette Speicher frei

25 zugänglich sein. Erst nach dem Endtest wird der Chip durch einen Transportcode gesichert. Danach ist der Zugriff auf den Kartenspeicher nur für berechtigte Stellen, die den Transportcode kennen, möglich. Ein Diebstahl fabrikneuer Halbleiter bleibt damit ohne Folgen. Berechtigte Stellen können Personalisierer bzw. Kartenausgeber sein. Für das Einbetten und Bedrucken sind keine weiteren Sicherungsfunktionen notwendig. Die betreffenden

30 Firmen brauchen den Transportcode nicht zu kennen.

Im allgemeinen überträgt nicht der Kartenhersteller, sondern die ausgebende Stelle (zum Beispiel Bank, Telefongesellschaft, Krankenkasse etc.) die personenspezifischen Daten in

die Karte. Diesen Vorgang nennt man Personalisierung. Für sie ist die Kenntnis des Transportcodes notwendig.

Das Ausgeben der Karte, also der Transport von der ausgebenden Stelle zum Karteninhaber, stellt ein weiteres Sicherheitsproblem dar. Genau genommen ist nur die persönliche Ausgabe an den Karteninhaber gegen Unterschrift und Vorlage des Personalausweises sicher. Ein Versand per Post ist zwar oft wirtschaftlicher, aber auch ziemlich unsicher. Ein Problem ist auch das Übermitteln der PIN an den Karteninhaber, hier muß die gleiche Sorgfalt wie für die Karte gelten.

10

Bedingt durch die brisanten, sicherheitsrelevanten Inhalte der auf Smartcard-Controllern befindlichen Speicher ist neben der Beachtung dieser Sicherungsmaßnahmen ein zusätzlicher Schutz gegen mögliche Aktivitäten von Hackern zu gewährleisten, die sich auf alle Phasen des Lebenslaufes einer Smartcard - beginnend von der Herstellung, über Transport, Nutzung der Karte bis zu Manipulationen unbrauchbar gewordener Karten - erstrecken.

15

Bei der Programmierung von größeren Mengen von Daten/Code in NV-Memories (zum Beispiel bei der Personalisierung in das EEPROM) entsteht ein relativ großer Zeitverlust einerseits durch den Datentransport via SFR-Bus, andererseits durch die notwendige Verifikation der geschriebenen EEPROM-Daten nach dem Programmieren jeder Page.

20

Zur Zeit bieten die Standard-Befehlssätze von Controllern für den Code-Memory-Bereich ausschließlich lesende Instruktionen. D.h., aus NV-Memories können Daten entweder als Instruction-Code abgerufen oder als Datenwort („Byte“) durch eine sog. MOVC-Instruktion gelesen werden.

25

Ein Beschreiben/Programmieren von Daten in das NV-Memory erfolgte bisher ausschließlich über den Registersatz des jeweiligen Memory-Interfaces, d.h., der Datenweg beim Beschreiben des NV-Memories ist komplett getrennt vom Datenweg des Code-Fetch / MOVC-Lesens.

30

Das Beschreiben erfordert mehrere Schreibzugriffe auf Memory-Interface-Register: Beschreiben der Adreß-Register für Page-Adresse und Byte-Adresse, Beschreiben des Daten-

Registers und des Kontroll-Registers.

Das bisherige Verfahren zum Beschreiben von NV-Memories ist gegenüber dem Code-Fetch/Lesen sehr langsam, da es je nach Zugriffsart zwei bis fünf Registerzugriffe pro geschriebenen Datenwort erfordert, während Code-Fetch und MOVC-Lesen im schnellen Code-Fetch-Takt des Prozessors ablaufen.

Da das Schreiben hierbei ausschließlich über die Register-Schnittstelle des Memory-Interfaces läuft, hat die Memory-Management-Unit, die das Mapping und die Zugriffsrechte des Code-Memories insgesamt kontrolliert, beim Beschreiben des NV-Memories keinen Einfluß. Daher kann das Beschreiben des Memories nur unter Kontrolle des Operating-Systems des Controllers geschehen und ist für Applikations-SW nur durch spezielle Calls auf System-Routinen möglich.

Der Erfindung liegt deshalb die Aufgabe zugrunde, ein Verfahren, eine Anordnung sowie ein entsprechendes Computerprogrammprodukt und ein entsprechendes computerlesbares Speichermedium der gattungsgemäßen Art anzugeben, durch welche die Nachteile der herkömmlichen Vorgehensweisen vermieden werden und durch welche es ermöglicht wird, in kürzestmöglicher Zeit Daten in ein NV-Memory zu schreiben, ohne wesentliche Eingriffe in bisher benutzten Verfahren vornehmen zu müssen, sowie einen höheren Schutz vor Programmierfehlern zu gewährleisten.

Erfindungsgemäß wird diese Aufgabe gelöst durch die Merkmale im kennzeichnenden Teil der Ansprüche 1, 12, 14 und 15 im Zusammenwirken mit den Merkmalen im Oberbegriff. Die Unteransprüche enthalten zweckmäßige Ausgestaltungen der Erfindung.

Ein besonderer Vorteil des Verfahrens zum Beschreiben von NV-Memories in einer Controller-Architektur besteht darin, daß (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Page-Adreß-Pointerregister des NV-Memories aktualisiert werden.

- Eine Anordnung zum Beschreiben von NV-Memories in einer Controller-Architektur ist vorteilhafterweise so eingerichtet, daß sie einen Prozessor umfaßt, der derart eingerichtet ist, daß ein Beschreiben von NV-Memories in einer Controller-Architektur durchführbar ist, wobei (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Page-Adreß-Pointerregister des NV-Memories aktualisiert werden.
- 10 Ein Computerprogrammprodukt zum Beschreiben von NV-Memories in einer Controller-Architektur umfaßt ein computerlesbares Speichermedium, auf dem ein Programm gespeichert ist, das es einem Computer oder Smartcard-Controller ermöglicht, nachdem es in den Speicher des Computers oder des Smartcard-Controllers geladen worden ist, ein Beschreiben von NV-Memories in einer Controller-Architektur durchzuführen, wobei (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Page-Adreß-Pointerregister des NV-Memories aktualisiert werden.
- 20 Um ein Beschreiben von NV-Memories in einer Controller-Architektur durchzuführen, wird vorteilhaft ein computerlesbares Speichermedium eingesetzt, auf dem ein Programm gespeichert ist, das es einem Computer oder Smartcard-Controller ermöglicht, nachdem es in den Speicher des Computers oder des Smartcard-Controllers geladen worden ist, das Beschreiben von NV-Memories in einer Controller-Architektur durchzuführen, wobei (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Page-Adreß-Pointerregister des NV-Memories aktualisiert werden.
- 30

Vorteilhaft wird ferner zum Beschreiben des NV-Memories der Befehlssatz des Controller-Cores um zusätzliche Move-Code-Write-Instruktionen (MOVCWR-Instruktionen) er-

weitere. In bevorzugter Ausgestaltung des erfindungsgemäßen Verfahrens ist vorgesehen, daß die zusätzlichen Instruktionen des Controller-Cores eine Übergabe der Parameter für Adreß-Pointer und für den zu schreibenden Datenwert oder das zu schreibende Datenwort durchführen und entsprechende Kontrollsignale für eine sog. Memory-Management-Unit (MMU) und NV-Memory-Interfaces aktivieren.

Als vorteilhaft erweist es sich, daß bei Vorhandensein einer Memory-Management-Unit (MMU) die Adreßverarbeitung für die MOVCWR-Instruktionen in gleicher Weise erfolgt wie die Verarbeitung von Code-Fetches oder MOVC-Instruktionen. Darüber hinaus ist in bevorzugter Ausgestaltung des erfindungsgemäßen Verfahrens vorgesehen, daß bei Vorhandensein einer Memory-Management-Unit (MMU) des Controllers diese MMU um einen Kontrollsignalpfad erweitert wird.

Vorteilhaft werden bei Vorhandensein einer MMU nur Adreßbereiche des NV-Memories beschrieben, die von der MMU freigegeben sind. Zum Beschreiben von NV-Memories in einer Controller-Architektur kann es sich als vorteilhaft erweisen, daß bei Vorhandensein einer MMU ein spezielles Mapping des Code Memories innerhalb des Adreßbereichs des Controllers berücksichtigt wird.

In weiterer bevorzugter Ausgestaltung des erfindungsgemäßen Verfahrens ist vorgesehen, daß nacheinander mehrere Datenwerte und/oder Datenworte mit derselben Pageadresse geschrieben werden.

Vorteilhaft wird durch Beschreiben des Control-Registers des NV-Memories der Inhalt des Cache-Page-Registers in das NV-Memory programmiert. Darüber hinaus ist in bevorzugter Ausgestaltung des erfindungsgemäßen Verfahrens vorgesehen, daß beim Wechsel auf eine neue Pageadresse bei einer MOVCWR-Instruktion das Cache-Page-Register des NV-Memories gelöscht wird.

Ein weiterer Vorteil des erfindungsgemäßen Verfahrens besteht darin, daß ein ungewolltes Programmieren alter Page-Register-Inhalte unter falscher Adresse verhindert wird. Darüber hinaus ist in bevorzugter Ausgestaltung der erfindungsgemäßen Anordnung vorgesehen, daß der Prozessor Teil eines Smartcard-Controllers und die Anordnung eine Smartcard ist.



Das erfindungsgemäße Verfahren bietet gegenüber dem bisher rein durch das Register-Interfaces des NV-Memories unterstützten Beschreiben des Cache-Pageregister mehrere Vorteile.

5

Das Beschreiben des NV-Memories mit MOVCWR erfordert pro Datenwort (Byte) nur eine MOVCWR-Instruktion mit Übergabe der beiden Parameter für den Adreßpointer und das Datenwort. Bei mehreren aufeinander folgenden MOVCWR-Instruktionen kann wie beim MOVC-Lesen ein „Autoincrement“ des Adreßpointers benutzt werden. Dieser

10 Befehlsaufruf stellt eine erhebliche Beschleunigung des Schreibvorganges gegenüber dem Schreiben via Adreß/Daten-Registersatz des NV-Memories da.

Spezielle Adreßbereichs-Mappings oder Zugriffs-Einschränkungen des Code-Memories, die von einer eventuell vorhandenen MMU überwacht werden, sind für MOVCWR auf  
15 gleiche Weise gültig wie für Code-Fetch und MOVC, d.h., der Prozessor-Core sieht bei der Ausführung von MOVCWR das gleiche Memory-Mapping wie bei Code-Fetch / MOVC.

Daher ist es auch einer Applikations-SW möglich, direkt die MOVCWR-Instruktion zu  
20 verwenden, um das Cache-Pageregister eines NV-Memories zu beschreiben, ohne einen System-Call aufrufen zu müssen. Die Kontrolle über die Zugriffsrechte auf das Memory behält das OS des Controllers über die Konfiguration der MMU Kontroll-Register.

Ein fehlerhaftes Programmieren alter Inhalte des Cache-Page-Registers eines NV-Memo-  
25 ries an eine falsche Pageadresse ist nicht mehr möglich, da das Cache-Page-Register mit jedem MOVCWR, dessen Adreß-Pointer die Pageadresse ändert, zurückgesetzt wird.

Die Erfindung wird nachfolgend in einem Ausführungsbeispiel näher erläutert.

30 Das vorgestellte Verfahren besteht in einer Erweiterung des Befehlssatzes des Controllers um sog. MOVCWR (move code write) Instruktionen, die es ermöglichen, ein definiertes Datenwort (Byte) an eine definierte Zieladresse innerhalb eines NV-Code-Memories zu schreiben. Das Datenwort (Byte) wird hierbei an die korrekte Position des Cache-Pagere-

gisters des jeweiligen NV-Memories geschrieben und die Pageadreß-Pointerregister des Memories mit der zugehörigen Pageadresse aktualisiert.

5 Wenn bei advanced Smartcard-Controllern eine MMU (Memory Management Unit) vorhanden ist, geschieht dieses MOVCWR-Schreiben in das Cache-Pageregister, wie das MOVC-Lesen bzw. der Code-Fetch, unter voller Kontrolle dieser MMU, so daß nur auf Adreßbereiche des Speichers geschrieben werden kann, die grundsätzlich von der MMU dafür freigegeben sind. Spezielles Mapping des Code Memories innerhalb des Adreßbereiches des Controllers wird hierbei berücksichtigt.

10

Auf diese Weise können nacheinander mehrere Bytes/Worte mit derselben Pageadresse geschrieben werden, um das Cache-Pageregister zu füllen. Durch Beschreiben des Control-Registers des jeweiligen NV-Memories kann dann der Inhalt des Cache-Page-Registers in das NV-Memory programmiert werden.

15

Jeder Wechsel auf einer neuen Pageadresse bei einer MOVCWR-Instruktion hat ein sofortiges Löschen des Cache-Pageregisters des NV-Memories zur Folge, um ein Programmieren von Daten unter der neuen Pageadresse zu ermöglichen und ein ungewolltes Programmieren alter Pageregister-Inhalte unter falscher Adresse zu verhindern.

20

In der beispielhaften Ausführungsform wird der Befehlssatz des Controller-Cores um zusätzliche MOVCWR-Instruktionen erweitert, um das Beschreiben von NV-Memories in erfindungsgemäßer Weise auszuführen. Die zusätzlichen MOVCWR-Instruktionen gewährleisten die eine Übergabe der Parameter für den Adreß-Pointer und den zu schreiben-  
25 den Datenwert und aktivieren entsprechende Kontrollsignale für MMU und Memory-Interfaces.

30

Eine eventuell vorhandene MMU (Memory Management Unit) des Controllers wird erweitert um einen entsprechenden Kontrollsignal-Pfad, der bei der Ausführung der MOVCWR Instruktion die entsprechenden Chip-Select-Signale für die Memory-Interfaces generiert. Die Adreßverarbeitung für die MOVCWR Instruktionen (bez. Mapping und Access Rights) unterscheidet sich hierbei nicht von der Verarbeitung von Code-Fetches oder MOVC-Instruktionen.

Die Memory-Interfaces der NV-Memories unterstützen diese Funktion durch einen entsprechenden Write-Mode für die Cache-Pageregister und einer Aktualisierungsfunktion der Adreß-Register nach jedem MOVCWR Vorgang. Außerdem führt eine Reset-Logik  
5 vor jedem MOVCWR-Vorgang einen Adreßvergleich zwischen alter und neuer Pageadresse durch und löst gegebenenfalls bei einem Adreßwechsel vor dem Beschreiben des Cache-Pageregisters ein Löschen des alten Registerinhaltes aus.

Die Erfindung ist nicht beschränkt auf die hier dargestellten Ausführungsbeispiele. Vielmehr ist es möglich, durch Kombination und Modifikation der genannten Mittel und  
10 Merkmale weitere Ausführungsvarianten zu realisieren, ohne den Rahmen der Erfindung zu verlassen.

PATENTANSPRÜCHE

1. Verfahren zum Beschreiben von NV-Memories in einer Controller-Architektur,  
dadurch gekennzeichnet, daß (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und  
5 die Pageadreib-Pointerregister des NV-Memories aktualisiert werden.
2. Verfahren nach Anspruch 1,  
dadurch gekennzeichnet, daß zum Beschreiben des NV-Memories der Befehlssatz des Controller-Cores um zusätzliche Move-Code-Write-Instruktionen (MOVCWR-Instruktionen)  
10 erweitert wird.
3. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß die zusätzlichen Instruktionen des Controller-Cores eine  
15 Übergabe der Parameter für Adreib-Pointer und für den zu schreibenden Datenwert oder das zu schreibende Datenwort durchführen und entsprechende Kontrollsignale für eine Memory-Management-Unit (MMU) und NV-Memory-Interfaces aktivieren.
4. Verfahren nach einem der vorhergehenden Ansprüche,  
20 dadurch gekennzeichnet, daß bei Vorhandensein einer Memory-Management-Unit (MMU) die Adreibverarbeitung für die MOVCWR-Instruktionen in gleicher Weise erfolgt wie die Verarbeitung von Code-Fetches oder MOVC-Instruktionen.
5. Verfahren nach einem der vorhergehenden Ansprüche,  
25 dadurch gekennzeichnet, daß bei Vorhandensein einer Memory-Management-Unit (MMU) des Controllers diese MMU um einen Kontrollsignal-Pfad erweitert wird.

6. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß bei Vorhandensein einer MMU nur Adreßbereiche des NV-Memories beschrieben werden, die von der MMU freigegeben sind.
- 5 7. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß bei Vorhandensein einer MMU ein spezielles Mapping des Code Memories innerhalb des Adreßbereichs des Controllers berücksichtigt wird.
8. Verfahren nach einem der vorhergehenden Ansprüche,  
10 dadurch gekennzeichnet, daß nacheinander mehrere Datenwerte und/oder Datenworte mit derselben Pageadresse geschrieben werden.
9. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß durch Beschreiben des Control-Registers des NV-Memories  
15 der Inhalt des Cache-Page-Registers in das NV-Memory programmiert wird.
10. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß beim Wechsel auf eine neue Pageadresse bei einer MOVCWR-Instruktion das Cache-Page-Register des NV-Memories gelöscht wird.  
20
11. Verfahren nach einem der vorhergehenden Ansprüche,  
dadurch gekennzeichnet, daß ein ungewolltes Programmieren alter Page-Register-Inhalte unter falscher Adresse verhindert wird.
- 25 12. Anordnung mit einem Prozessor, der derart eingerichtet ist, daß ein Beschreiben von NV-Memories in einer Controller-Architektur durchführbar ist, wobei (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Pageadreß-Pointerregister des NV-Memories  
30 aktualisiert werden.

13. Anordnung mit einem Prozessor nach Anspruch 12,  
dadurch gekennzeichnet, daß der Prozessor Teil eines Smartcard-Controllers und die An-  
ordnung eine Smartcard ist.
- 5 14. Computerprogrammprodukt, das ein computerlesbares Speichermedium umfaßt,  
auf dem ein Programm gespeichert ist, das es einem Computer oder Smartcard-Controller  
ermöglicht, nachdem es in den Speicher des Computers oder des Smartcard-Controllers  
geladen worden ist, ein Beschreiben von NV-Memories in einer Controller-Architektur  
durchzuführen, wobei (ein) definierte(r) Datenwert(e) oder (ein) definierte(s) Daten-  
10 wort(e) an (eine) definierte Zieladresse(n) innerhalb des NV-Memories geschrieben werden  
(wird), indem die (der) Datenwert(e) bzw. die (das) Datenwort(e) an die vorgegebene Po-  
sition des Cache-Pageregisters des NV-Memories geschrieben werden (wird) und die Page-  
adreß-Pointerregister des NV-Memories aktualisiert werden.
- 15 15. Computerlesbares Speichermedium, auf dem ein Programm gespeichert ist, das es  
einem Computer oder Smartcard-Controller ermöglicht, nachdem es in den Speicher des  
Computers oder des Smartcard-Controllers geladen worden ist, ein Beschreiben von NV-  
Memories in einer Controller-Architektur durchzuführen, wobei (ein) definierte(r) Da-  
tenwert(e) oder (ein) definierte(s) Datenwort(e) an (eine) definierte Zieladresse(n) inner-  
20 halb des NV-Memories geschrieben werden (wird), indem die (der) Datenwert(e) bzw. die  
(das) Datenwort(e) an die vorgegebene Position des Cache-Pageregisters des NV-Memories  
geschrieben werden (wird) und die Pageadreß-Pointerregister des NV-Memories aktuali-  
siert werden.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**